

基于 PLC 多层架构构建数控语言的方法与实现

董 禄, 严 义*, 邬惠峰

(杭州电子科技大学 计算机学院, 浙江 杭州 310018)

摘要:针对“使用梯形图等编程方式编写 PLC 运动控制程序,在调试和优化过程中,需要不断修改 PLC 控制程序和调整参数而导致开发效率低”的问题,提出了一种在 PLC 上构建数控语言的方法。详细介绍了构建数控语言的实现过程,包括数控指令的设计与实现、数控程序的编译和执行;重点说明了如何采用 PLC 模块化编程方式实现数控指令的功能,同时给出了数控程序解析执行控制算法;最后在可定制嵌入式 PLC 上构建了数控语言层,并通过具体实例进行了验证分析。研究表明,利用该方法能够方便地进行运动控制序列的调整和参数的设置,并且可以有效减少 PLC 控制程序的代码量,从而提高开发效率;此外,还可针对不同的实际应用,快速定制出相应的数控语言,满足灵活多变的控制要求,适合在工业生产中使用。

关键词:PLC; 数控语言; 梯形图; 运动控制

中图分类号:TP311; TP332.3; TH164

文献标志码:A

文章编号:1001-4551(2012)07-0856-05

Method and implementation of constructing numerical control language based on PLC multi-layer architecture

DONG Lu, YAN Yi, WU Hui-feng

(College of Computer, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: In the PLC with motion control functions, it is necessary to modify the PLC control programs and adjust the parameters constantly for motion control programs written with ladder diagram in debugging and optimization process, which will lead to the problem of low development efficiency. Aiming at this problem, a method to construct the numerical control language on the PLC was proposed. The implementation process to construct numerical control language was described in detail, including the design and implementation of the numerical control instructions, the compilation and execution of the numerical control programs. The way to implement the functions of numerical control instructions with PLC modular programming method was stressed, then the resolution and execution control algorithms of the numerical control programs were given out. At last, a specific example of building numerical control language layer on the customized embedded PLC was presented. The results indicate that the approach can conveniently adjust the sequence of motion control and setting parameters, which can effectively reduce the number of PLC control codes and improve the developing efficiency greatly. Besides, the method can quickly customize the corresponding numerical control language according to the different applications, so that it can meet the requirements of flexible control needs, and be suitable for industrial control.

Key words: programmable logic controller (PLC); numerical control language; ladder diagram; motion control

0 引 言

经过几十年的发展,PLC^[1]已经具备各种丰富的控制功能,远远超出了其传统的逻辑控制范围。目前,大多数 PLC 已具备了运动控制能力^[2-5],如:定位控

制、直线插补、圆弧插补等。具有运动控制功能的 PLC 在组成架构上主要有 2 种实现方式:①PLC 中内嵌运动控制单元,控制单元与 PLC 使用共同的处理器;② PLC 扩展外置运动控制单元,控制单元使用独立的处理器,通过总线与 PLC 通信。内嵌运动控制单元的

收稿日期:2012-02-03

作者简介:董 禄(1985-),男,湖北阳新人,主要从事嵌入式与智能控制技术方面的研究。E-mail:updee@yahoo.cn

通信联系人:严 义,男,教授,硕士生导师。E-mail:yybjyyj@163.com

PLC 通常采用梯形图编程或结构化文本方式编程;外置运动控制单元通常有独立的编程语言,如 G 代码语言^[6]等。为了统一 PLC 的运动控制编程,PLCopen 组织^[7]制定了运动控制编程标准,提出使用运动控制功能块(PLCopen Motion Control Function Blocks)来编程,该标准以 IEC61131-3 中功能块的概念为基础,通过 PLC 编程方式来进行运动控制程序的设计,使运动控制编程和逻辑控制编程在 PLC 中无缝集成^[8],减少了开发时间和成本^[9]。然而,使用采用 PLC 编程语言(如梯形图)进行编程,在运动轨迹不固定、控制过程复杂的情况下,将导致很大的代码量;并且无论任何改动都需要重写或更改控制程序,再重新下载执行,不仅调试过程繁琐,而且加重了维护和优化工作。虽然通过扩展使用额外的具有独立编程语言的运动控制单元,可以在程序开发过程中提高效率,但同时增加了系统的复杂性以及开发成本。

针对上述问题,本研究提出:在内嵌运动控制单元的定制嵌入式 PLC 中增加适合运动控制的数控语言层,以实现运动控制序列和参数的调整,由此减少开发难度,提高开发效率。

1 PLC 构建数控语言的设计思想

1.1 PLC 的多层架构

本研究通过在现有的设计环境之上构建数控语言层,实现运动控制序列和参数的调整,以有效减少 PLC 控制程序代码量。构建数控语言层的 PLC 多层架构的层次如图 1 所示。各层次均以其下层为基础,并为其上层提供服务。

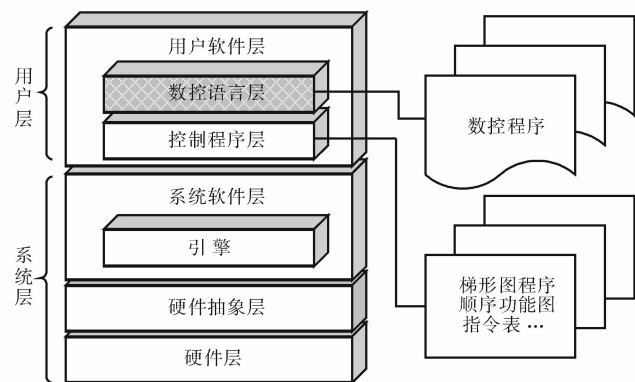


图1 构建数控语言层的 PLC 多层架构

系统架构层次主要由系统层和用户层两大部分组成,下面以可定制嵌入式 PLC 系统进行阐述。

(1)系统层。由硬件层、硬件抽象层和引擎构成,其各部分功能与实现描述如下:

① 硬件层,以 SOC 芯片为核心,加上外围电路和输入、输出模块构成了 PLC 的硬件系统;

② 硬件抽象层,实现硬件的驱动和抽象,为上层引擎提供了硬件访问的统一接口(硬件抽象层由 C 语言和汇编语言开发实现);

③ 引擎,运行在硬件抽象层之上,是 PLC 的执行系统,主要实现中断、定时器、运动控制算法、数控程序解析执行算法及其他各类算法和用户程序的调用(引擎程序使用 C 语言开发实现)。

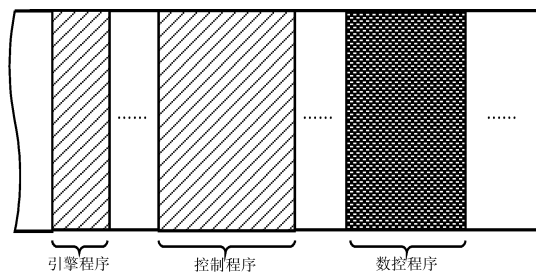
(2)用户层。由控制程序层和数控语言层构成,其功能与实现描述如下:

① 控制程序层,采用符合 IEC61131-3 标准^[10]的编程语言,如梯形图(LD)、顺序功能图(SFC)等,实现用户控制程序的开发(该方法实现的控制程序采用梯形图编程语言开发);

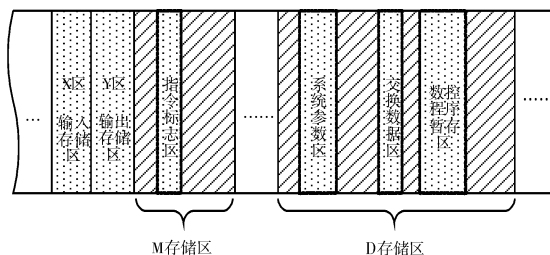
② 数控语言层,建立在控制程序层之上,采用数控语言的形式实现对控制系统功能的定制,主要是对运动控制序列和参数的定制。

1.2 存储空间分配

PLC 控制功能主要由各类程序实现,包括引擎、用户控制程序和数控程序。在 Flash 存储空间中,需划分 3 个区域分别用来存放引擎程序、用户控制程序和数控程序,其空间分配如图 2(a)所示。各类程序存放的位置固定,程序通过地址调用的方式来执行。



(a) Flash 空间分配



(b) RAM 空间分配

图2 PLC 中专用存储空间分配

本研究在 RAM 内存空间的数据存储区 D 区中,划分出 3 个专用数据区,其空间分配如图 2(b)所示。

其中,数控程序暂存区用于暂存调试或运行中的数控指令序列;交换数据区存放指令执行过程中的指令参数;系统参数区为运动控制参数设定区域,是引擎中运动控制算法使用的专用存储区域。在 M 区划分出指令标志区,作为用户控制程序中梯形图代码块的执行标志。

2 数控语言的实现

数控语言用来描述加工零件的几何形状、几何元素间的相互关系及加工过程、工艺参数等^[11]。

2.1 数控指令的设计

数控指令描述了指令所具备的功能和所需的参数,指令格式采用指令名与参数的结合方式,即:

指令名 参数 1, 参数 2, 参数 3, ..., 参数 n

指令名用来描述指令完成的功能,一般采用易于理解和记忆的符号组成,如 XLS(X-Axis Ladder parameter Set)表示 X 轴梯形加减速运动参数设置, XLM(X-Axis Ladder Movement)表示 X 轴梯形加减速运动, DELAY 表示延时, END 表示程序结束;根据需求,也可设计成 G 代码指令形式。参数为实现功能所需的数据集合,为了在 PLC 引擎中方便处理,本研究约定指令码存储空间为 1 个字节,参数存储空间均为 4 个字节。指令名和对应的指令码具有唯一性,指令设计示例如表 1 所示。

表 1 数控指令设计示例

指令码	指令名	参数个数	参数 1	参数 2	参数 3
0	XLS	3	4B	4B	4B
1	XLM	3	4B	4B	4B
2	DELAY	1	4B	-	-
100	END	0	-	-	-

所有与具体应用相关的指令形成指令集。本研究通过约定指令间的使用规则,形成一套具有针对性的数控语言。如在表 1 中,约定指令 XLS 必须在指令 XLM 之前使用;约定数控程序必须以 END 指令结束。

2.2 数控指令的实现

运动控制和逻辑控制相关数控指令(如 XLM),其功能通过梯形图编程实现。梯形图模块化实现单个指令的功能,包括传递参数、设置参数、启动运动控制命令和指令标志复位等。

指令 XLS、XLM 及 DELAY 的梯形图实现如图 3 所示。图 3 中常开触点 M2000、M2001 和 M2002 分别为 XLS、XLM、DELAY 的指令标志位,该标志位与指令

码具有一一对应关系。指令标志位右侧串联的梯形图代码块实现数控指令的功能。标志位的使能过程由引擎中数控程序解析执行算法控制,该算法负责数控指令的解析和数控程序的执行;VD6000 为交换数据区起始地址;VD510、VD430 和 VD440 分别是 X 轴初始速度、加速度和减速度的参数设置地址,VD2200、VD2150 和 VD2100 分别为运动距离、运动速度和运动方向设置地址,以上数据地址处于系统参数区;M20 是梯形加减速曲线运动命令启动标志,启动后由运动控制算法完成相关计算和脉冲输出等操作。T0 为 1 ms 定时器,由 VD6000 设置定时时间,定时完成后将指令标志 M2002 复位。

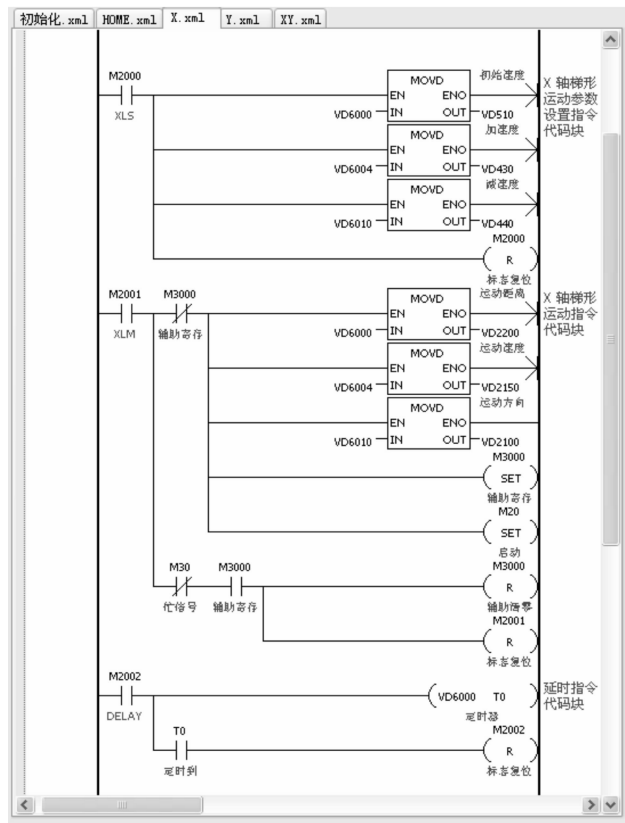


图 3 数控指令的梯形图实现

由图 3 可知,实现指令功能的代码块具有独立性。本研究通过在梯形图代码块末尾将指令标志位复位,使其恢复初始状态,以实现代码块的复用。

2.3 数控程序的编译

编译过程中系统首先判断指令名的合法性,将合法指令名转换成指令码。然后将参数进行转换处理,如指令 XLM 500 mm, 60 mm/s, 0, 表示 X 轴以 60 mm/s 的目标速度正向(0 为正向, 1 为反向)移动 500 mm, 由于 PLC 中使用脉冲量进行运动控制,需要将指令转换成脉冲量形式。本研究设脉冲当量为 1 μm , 上述指

令运动距离 500 mm 将被转换成 500 000 pulse,运动速度 60 mm/s 将被转换成 60 000 pulse/s。

通过上述步骤,对如下的指令序列进行编译转换:

N101 XLS 10 mm/s, 50 ms, 30 ms

N102 XLM 500 mm, 60 mm/s, 0

N103 DELAY 2 s

其中,N101 行表示将梯形加减速曲线运动初始速度设置为 10 mm/s,加速度和减速度时间分别设置为 50 ms、30 ms;N103 行表示延时 2 s,由于梯形图中使用 1 ms 定时器,需进行参数转换。以上代码转换成目标格式的数据如下:

0, 10 000, 50, 30

1, 500 000, 60 000, 0

2, 2 000

数控程序中指令经编译转换后,按指令先后顺序组合,形成以二进制数据保存的目标文件。本研究通过将编译后的程序下载到数控程序暂存区调试、执行,最终把调试好的数控程序写入 Flash 中的数控程序区保存,待以后使用时根据需要调出执行。

2.4 数控程序的执行

数控程序执行过程由引擎中数控程序解析执行算法控制。本研究设数控程序暂存区的起始地址为 VDaddrNC,交换数据区起始地址为 VDaddr,指令标志区起始地址为 VMaddr,指令解析表描述如表 2 所示。

表 2 数控指令解析表

指令码	参数个数	指令标志区索引地址
0	a	VMaddr
...	...	...
8	b	VMaddr + 1
...	...	...
16	c	VMaddr + 2
...	...	...
$8 \times n$	d	VMaddr + n

本研究设指令个数为 N ,在引擎中可将指令解析表通过 C 语言描述为:

```
unsigned short InstructionTable[N][3] =
{ {0, a, VMaddr}, ..., {8, b, VMaddr + 1}, ...,
  {16, c, VMaddr + 2}, ..., {8 × (N-1), d, VMaddr + N-1} };
```

数控程序解析执行算法通过查询指令解析表,完成指令解析执行,包含以下 5 个操作步骤:

(1)指令读取。从数控程序区当前指针位置读入一个字节数据,作为指令码,判断其是否为结束指令,若是,则结束;若不是,则进行步骤(2)。

(2)指令解析。根据得到的指令码,查询指令解

析表中对应的解析表项,记录当前指令的信息。

(3)数据传送。根据指令解析获取的参数个数和指令标志区索引地址,将参数传送到参数交换区,传入交换数据区的参数始终以 VDaddr 开始存放。

(4)启动执行。根据启动标志起始地址和指令码,将数控指令标志区中相应位置位,执行与数控指令对应的梯形图代码块。

(5)等待运行结束。以当前指令标志是否复位,作为指令执行完毕的依据。当前指令执行完毕后,若有外部停止信号,则结束;否则返回步骤(1)执行。

以上步骤完成数控指令的解析执行,数控程序执行流程如图 4 所示。

在实际设计中,本研究可设置一个输入点(如 X0)作为数控程序的启动开关。当本研究启动开关置位后,算法逐条解析存放在数控程序暂存区中的指令,将指令参数从数控程序暂存区传送至交换数据区,如上述 XLS、XLM 指令的设置交换数据区是以 VD6000 为起始地址,在执行过程中,系统都将指令参数按顺序传送至以 VD6000 起始的数据区;最后将梯形图中指令标志置位,如 M2000,执行梯形图程序,实现相应操作。

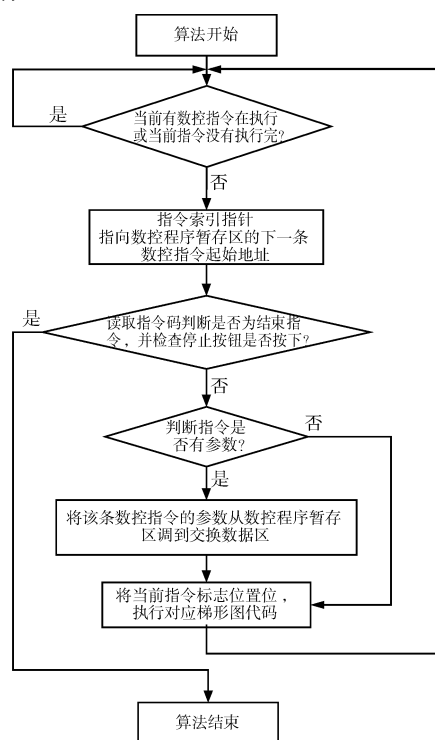


图 4 数控程序执行流程

3 实例分析

为验证方法的可行性和有效性,本研究将该构建

方法在自行开发的可定制嵌入式 PLC 上实现,并用于电感绕线机的控制,执行部件使用伺服电机。

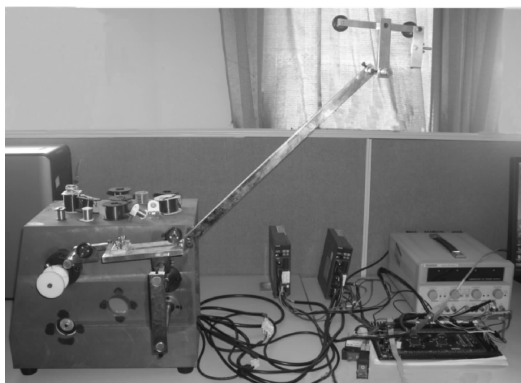


图 5 实物连接图

试验使用的实物连接图如图 5 所示。绕线机采用 X 轴和 Y 轴 2 轴控制,X 轴为旋转的线轴,Y 轴为通过丝杆做往返运动的排线轴。本研究根据绕线机的结构特征及工作原理,设计绕线机数控指令,所设计的部分关键指令如表 3 所示。通过自行研发的 PLC 开发环境 CASS 控制器算法平台,使用梯形图设计控制程序。

表 3 绕线机数控指令

指令名	指令含义	指令名	指令含义
X_HOME	X 回零点	Y_HOME	Y 回零点
XP	X 定位	YP	Y 定位
X_LINE	X 直线运动	Y_LINE	Y 直线运动
XY_LN	XY 同步直线	XY_LD	XY 同步梯形
	匀速运动		曲线运动
START	程序开始	END	程序结束

本研究使用 C#开发的数控程序编译器,实现数控程序的编译、下载、运行监控和指令编辑等功能,其界面如图 6 所示。

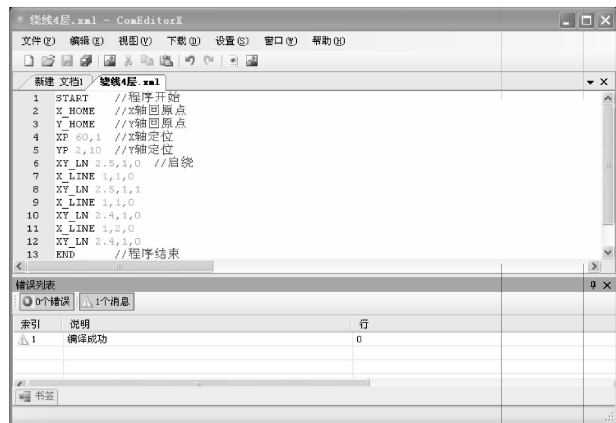


图 6 数控程序编译器

本研究通过编写绕线程序并联机实验,结果显示绕线机的绕制效果良好。由此可见,通过采用数控语言编写程序的方式,能方便地进行运动控制序列和参

数的修改,提高了开发效率。

4 结束语

通过对现有内嵌运动控制功能 PLC 进行分析的基础上,本研究提出了一种构建数控语言的方法来进行运动控制序列的改变和参数设置,并详细说明了其实现过程,最后通过具体应用实例进行了验证。研究结果表明:该方法具有实现简单、生成代码量小,并可针对不同的应用快速定制出相应的数控语言等优点,具有较好的实用价值。

参考文献 (References):

- [1] MORRIS S B. Programmable Logic Controllers [M]. Beijing: China Machine Press, 2006.
- [2] 严义, 朱旭燕. 基于 DSP 的 PLC 运动功能的研究[J]. 机电工程, 2011, 28(7): 818-822.
- [3] 汪小澄, 袁立宏, 张世荣. 可编程逻辑控制器运动控制技术[M]. 北京: 机械工业出版社, 2006.
- [4] ZHAO T, OSSEYRAN A. The Benefits of Integrating Motion Control into PLC System [C]//Proceedings of 2nd International Conference on Machine Automation. Finland: [s. n.], 1998: 29-41.
- [5] LI Wu. Study of multipoint position and reciprocating motion control based on PLC [J]. Microcomputer Information, 2008, 24(7): 52-54.
- [6] ISO 6983/1: Numerical control of machines - Program format and definition of address words - Part 1: Data format for positioning, line motion and contouring control systems [S]. ISO, 1982.
- [7] 彭瑜. 工控编程语言 IEC61131-3 和 PLCopen 组织[J]. 电气时代, 2006, 3(5): 32-36.
- [8] SÜNDER C, ZOITL A, MEHOFER F, et al. Advance use of PLCopen motion control library for autonomous servo drives in IEC61499 based automation and control systems [J]. Elektrotechnik and Informationstechnik, 2006, 123(5): 191-196.
- [9] MASSEY N, ENGINEER S. B&R Industrial Automation Corp. PLCopen motion control: reducing development time and cost with standardized motion programming [EB/OL]. [2008-05-19]. <http://www.e-driveonline.com/whitepapers/B&R2.pdf>.
- [10] JOHN K H, TIEGELKAMP M. Programming Industrial Automation Systems IEC 61131-3 [M]. Berlin: Springer-Verlag, 2001.
- [11] 黄文生, 张建新. 数控技术应用及数控系统开发 [M]. 北京: 国防工业出版社, 2009.

[编辑: 李 辉]